National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

Project Integration Architecture

http://www.grc.nasa.gov/WWW/price000/index.html

Ms. Theresa L. Benyo — Program Management
Theresa.L.Benyo@nasa.gov

Mr. Richard A. Blech — Program Management
Richard.A.Blech@nasa.gov — Chief, Engine Systems Branch

Dr. William H. Jones — Technical Lead
William.H.Jones-1@nasa.gov

Project Integration Architecture (PIA) Oversimplified Nutshell

Project Integration Architecture (PIA) is a distributed, object-oriented, architectural framework that provides (in a machine- and human-intelligible manner) for the generation, organization, publication, integration, consumption, and storage of all information involved in any process.

Project Integration Architecture (PIA) Synopsis

PIA is an object-oriented wrapping architecture that captures and encapsulates the entirety of any process.

1. All sources and sinks of information are conformed to a single, self-revealing architecture. Common tool sets are enabled.

2. The complete trail and structure of investigation is captured producing an auditable record. Notations, logs, journals and the like may also be captured.

3. Through the technologies of self-revelation and semantic infusion through class derivation, applications may interact with each other to provide for automatic transfer of information. The *n-squared* integration problem is eliminated and automated teaming is enabled.

4. Because information is “self-aware” through its encapsulation in an appropriate object, it can provide functionality appropriate for its kind. For example, dimensional information understands its own dimensionality and automatically converts to the appropriate system when accessed. Further, dimensional information disallows all accesses that are not dimensionally sensitive.

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

Key Object-Oriented Technologies Exploited by PIA

1. Self-revelation: Objects can reveal their kind and the extent of their content.

   1.1 Self-revelation of kind sets the expectation for the nature and content of an object.

   1.2 Self-revelation of content exposes the extent to which expectations are, in fact, fulfilled.

2. Semantic infusion through class derivation: By progressive derivation from parent to child, the nature of the derived child can be progressively defined.

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

CORBA::Object	A CORBA-Served instance
GObject	The self-revealing patriarch
GObjSta	Having boolean characteristics
GObjLck	Lockable (multi-accessor safe)
GObjDgn	Able to participate in a directed graph
GacBObj	Part of an application (access controls, signatures)
GacPara	A parameter of an application
GacParaArr	Array organization
GacParaArrDoub	Atomic Type: double
GacParaDimArr	Dimensional in nature
GacParaDimArrMfl	The dimensionality of massflow
GacParaGasArrMfl	A gas massflow
GacParaGasArrMflCr	A corrected massflow
GacParaGas1DGridMflCr	A 1-dimensional grid
LapPar1DGridMflCr	A 1-dimensional corrected massflow grid produced by Lapin

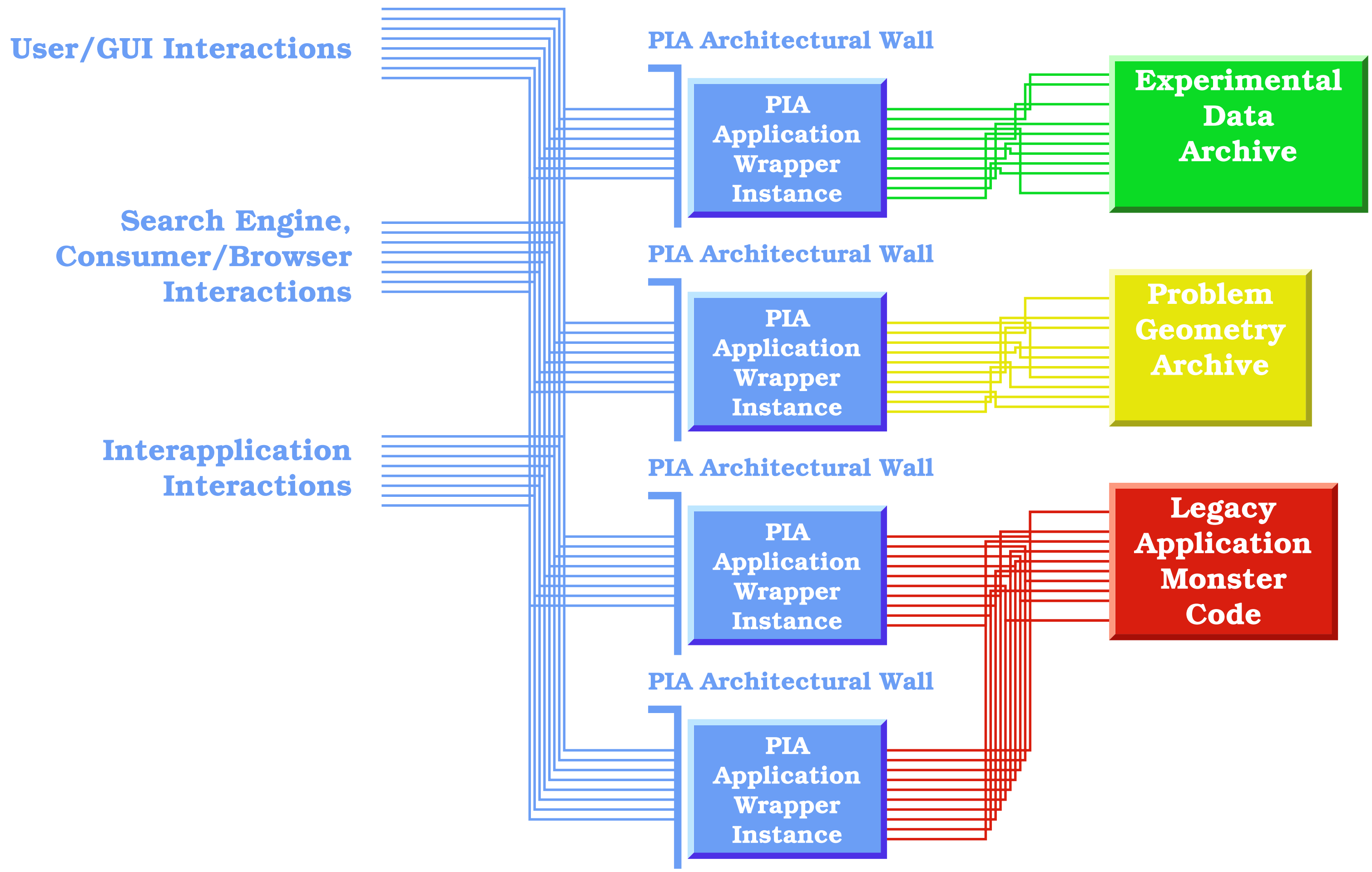
Ecdysiastical analysis

Semantic Infusion Through Class Derivation

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

Unclassified



PIA Application Architectural Wall Concept

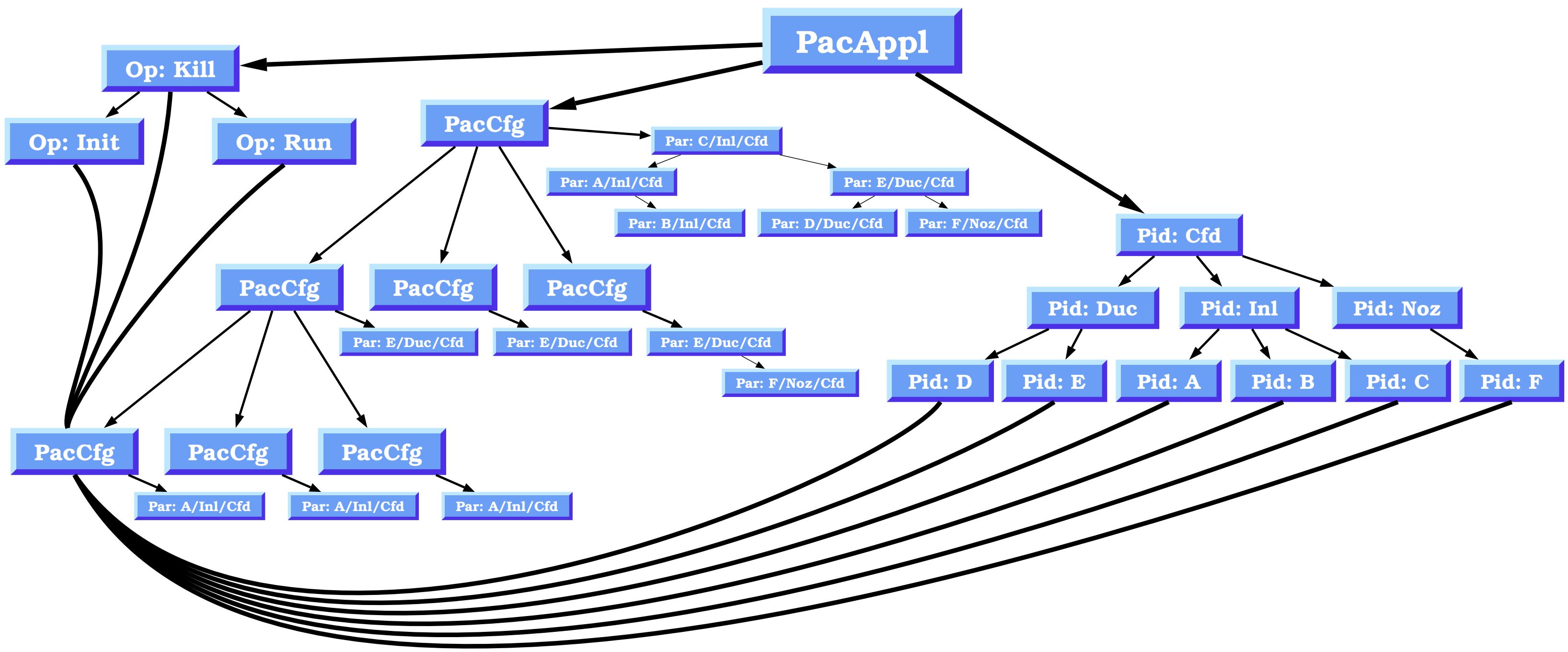
National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

Propulsion Systems Division

Engine Systems Branch

Unclassified



The PIA Self-Revealing Application Architecture

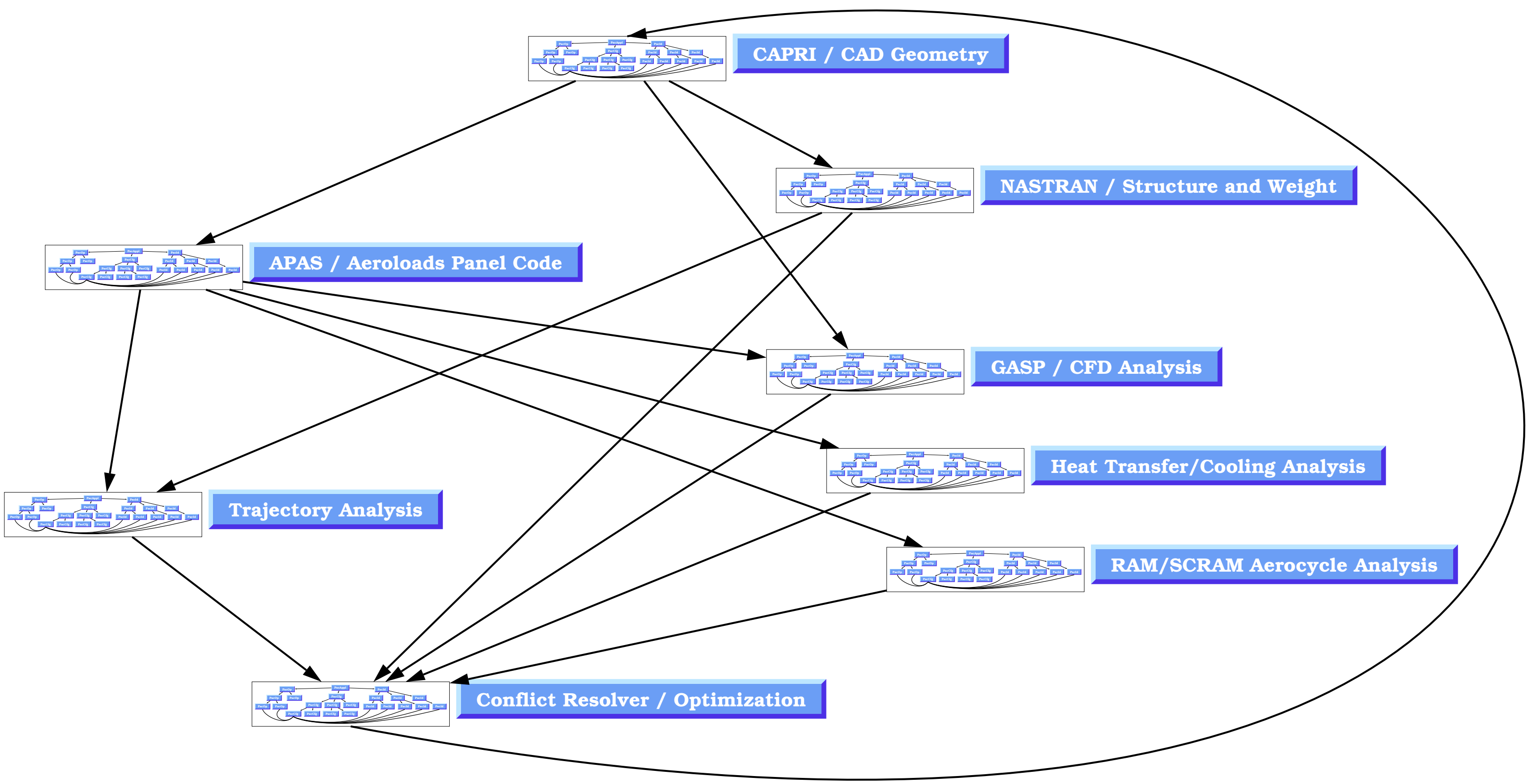
Propulsion Systems Division

Engine Systems Branch

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

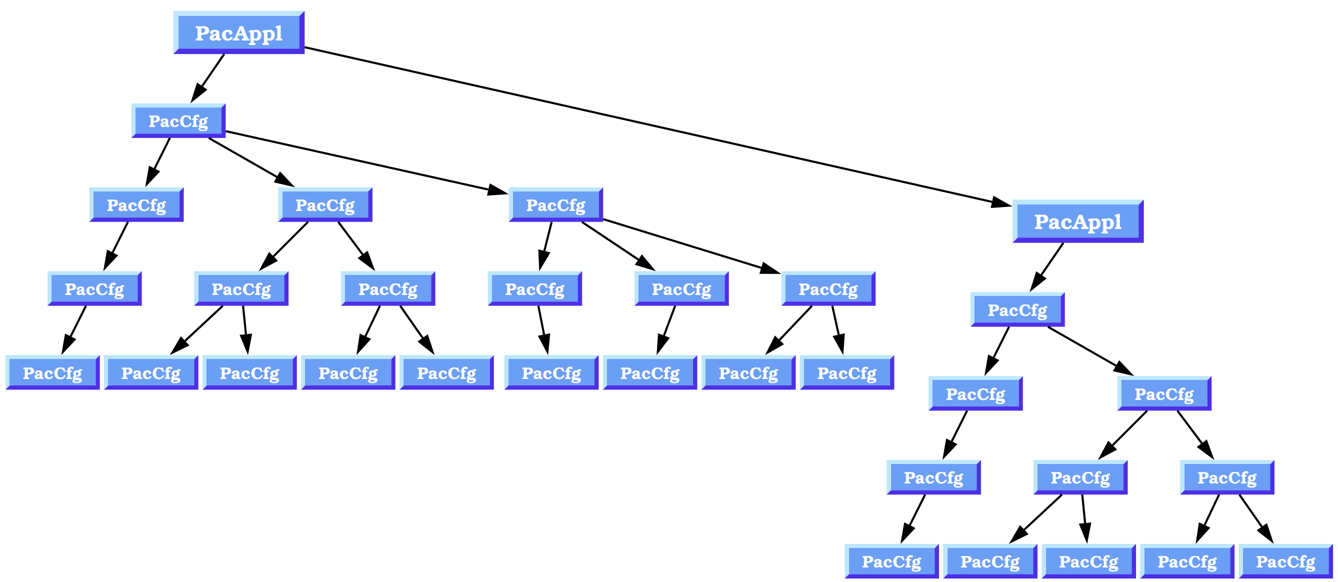
Unclassified



Integrated Application Graphs

National Aeronautics and
Space Administration

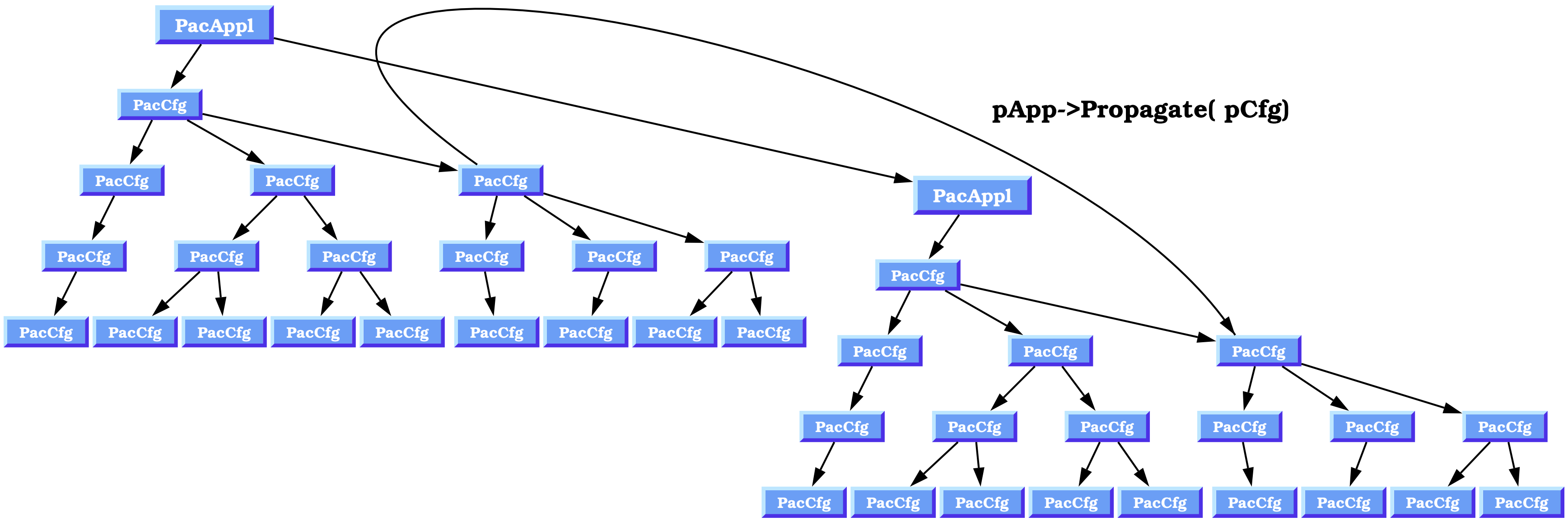
John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191



Data Configuration Prior to Propagation

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191



Data Configurations After Propagation has Occurred

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

Autonomous Solution Systems

Given a sufficently rich environment, PIA provides the basis upon which application graphs may be autonomously assembled.

1. The application object can reveal the set of parameters which it produces as output and, upon further interrogation, the input parameters needed to generate any particular output.

2. Given a desired output, an automated process similar to a program linker can identify the application producing that parameter. The needed inputs are noted and a recursive search performed to satisfy those needs until only inputs that can be regarded as an independent design vector remain.

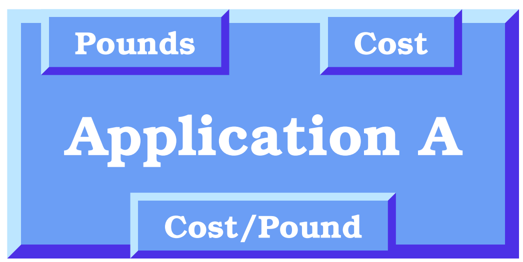
3. The assembled application graph can then be exercised by an “optimization” automaton that will, typically,

   3.1 Establish the sensitivity of the various elements of the independent design vector,

   3.2 Perform a genetic (or other) analysis to identify likely areas for global optima,

   3.3 Perform a true optimization operation, and

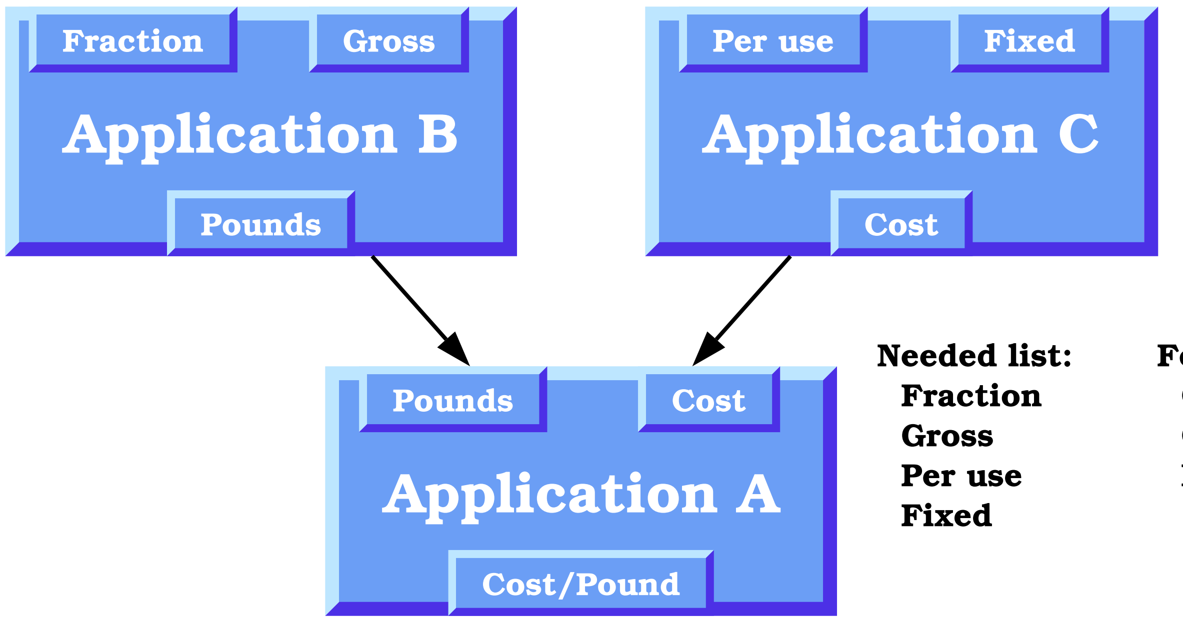
   3.4 Apply “six-sigma” technologies to enhance design reliability.



Needed list:
Cost
Pounds

Found List:
Cost/Pound

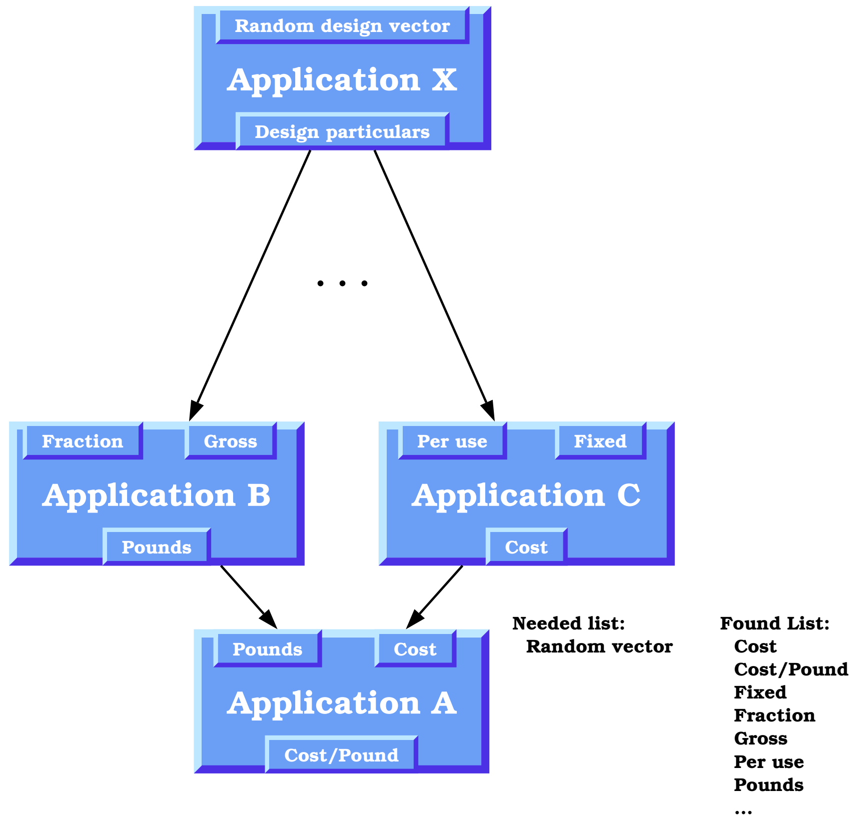
Continuing the Autonomous Assembly of an Application Graph



Needed list:
- Fraction
- Gross
- Per use
- Fixed

Found List:
- Cost
- Cost/Pound
- Pounds

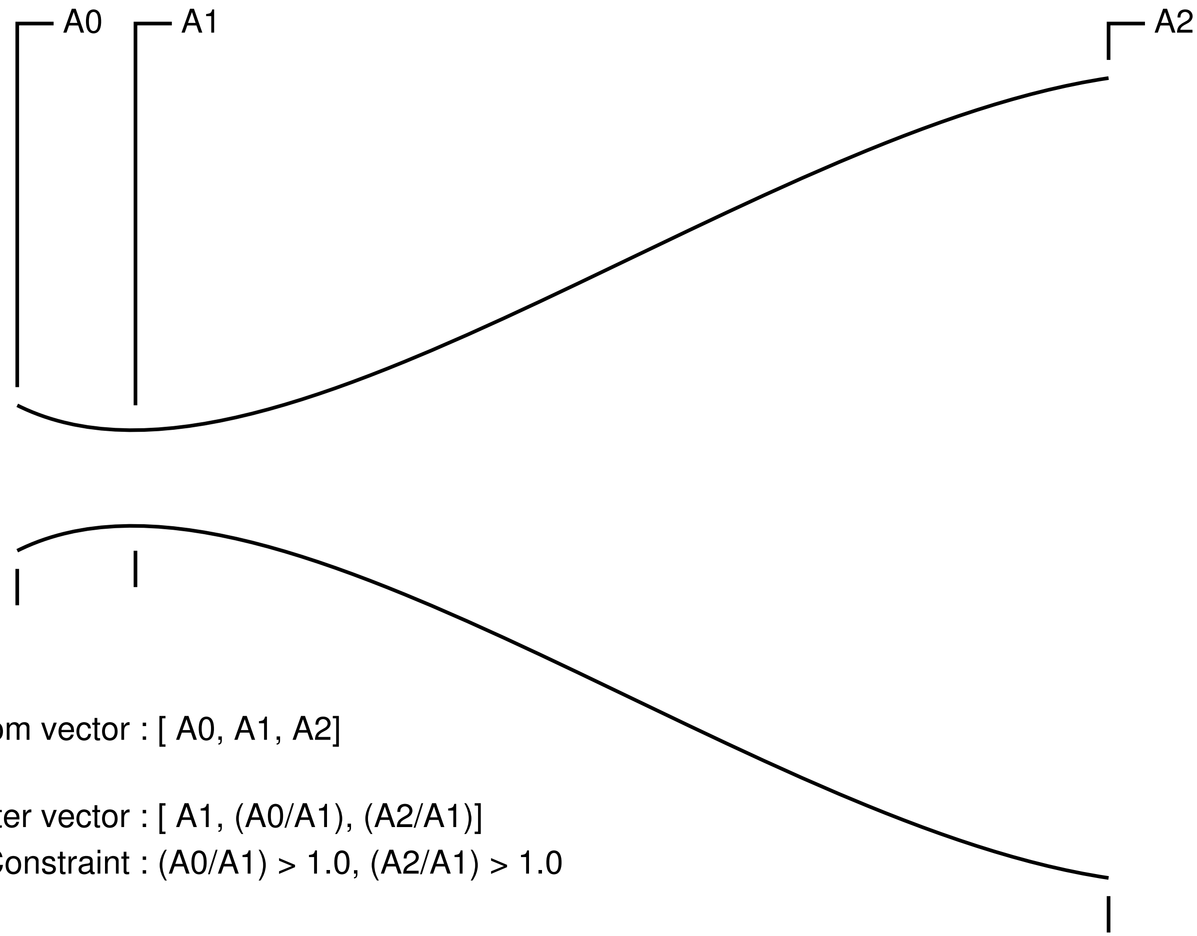
Further Recursion of the Autonomous Assembly Algorithm



Reduction to Applications Requiring Only Random Inputs

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191



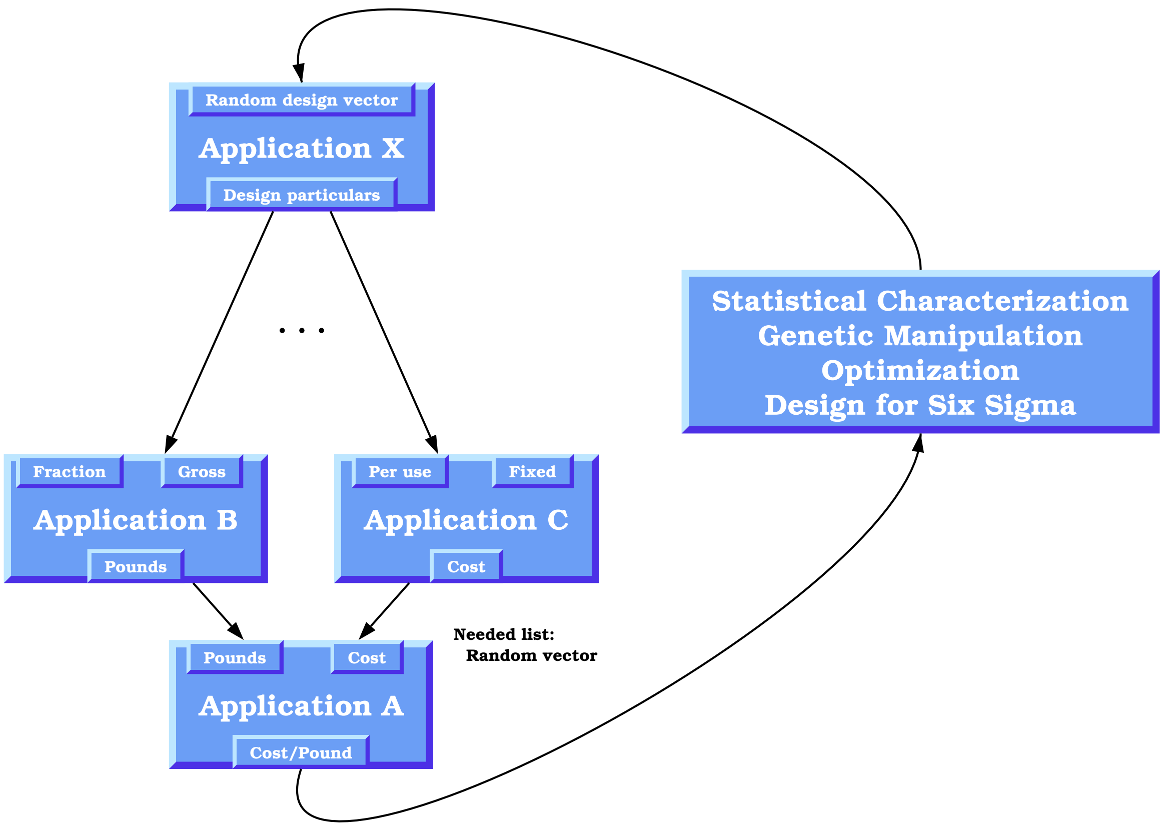
Random vector : [A0, A1, A2]

Better vector : [A1, (A0/A1), (A2/A1)]
Constraint : (A0/A1) > 1.0, (A2/A1) > 1.0

A Rocket Motor Design Application with Random Inputs

National Aeronautics and
Space Administration

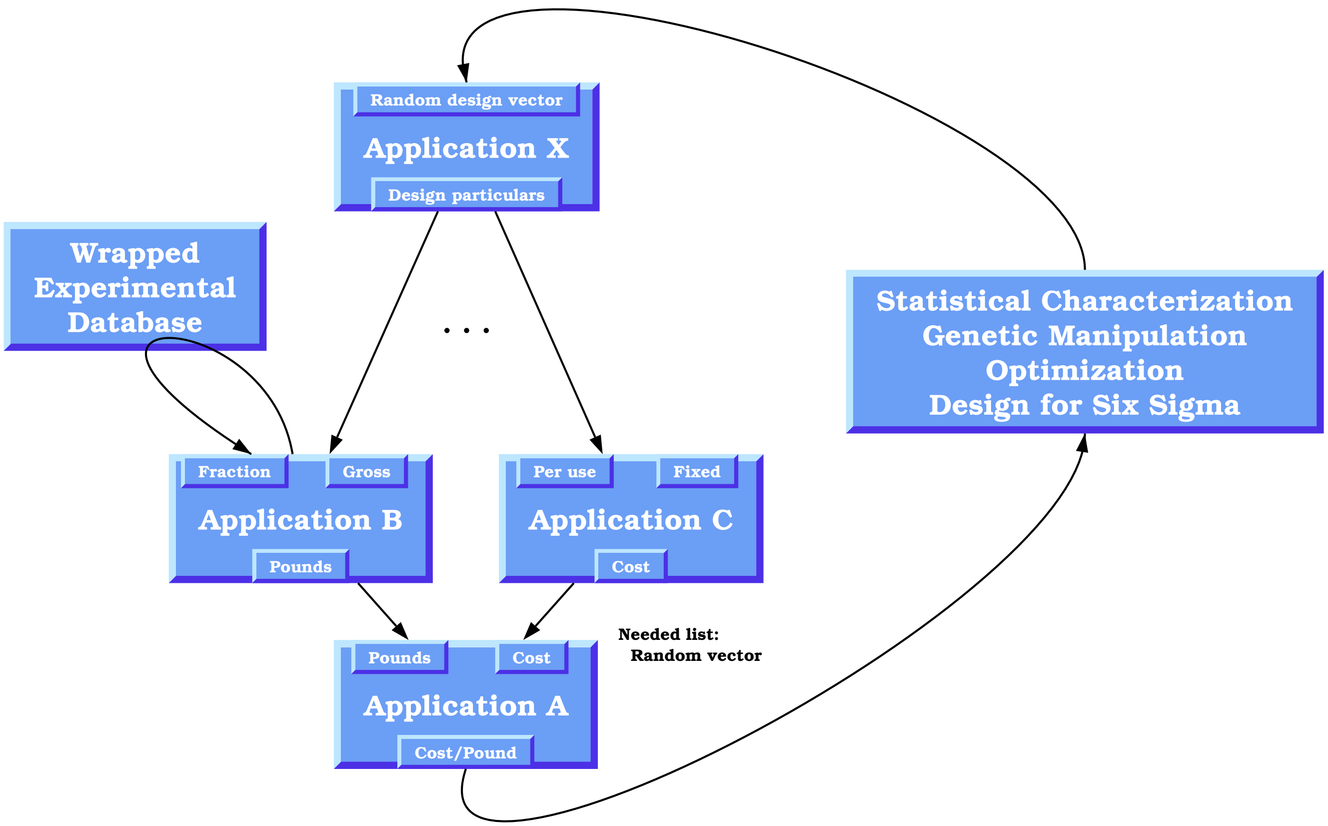
John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191



Application of Solution Initialization and Improvement Technology

National Aeronautics and Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191



Use of Relevant Experimental (or Other) Information

Automatic Solution System Benefits

1. Extension of application integration beyond the limits of human capability.
   1.1 With a 99.99% connection accuracy, an integration with 20,000 connections has an 86% chance of having one wrong connection.

2. Elimination of human failings from the solution formulation process.
   2.1 Accurate transfers of information.
   2.2 Dispassionate consideration of alternative strategies; one NASA, one collective.

3. Automated risk assessment of the solution process.
   3.1 Identification of weak or missing technology areas.

4. Discipline expert's team participation time reduced; time freed for discipline development.

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

Automatic Solution System Near-Term Demonstration

1. Automatic development and use of the quadratic formula.
   1.1 Develop the quadratic formula as a solution to a posed quadratic equation using basic axioms.
   1.2 Upon re-posing a quadratic equation problem, observe that the now-developed quadratic equation is used to solve the problem directly.
2. Quadratic formula problem chosen for its limited size.
3. Algorithm works in opposite direction: forward from a given start point rather than backward from a desired end result.
4. Algorithm would represent a step beyond current automatic theorem provers which require an asserted end point to work toward.
5. Direct use of the developed theorem for the second problem demonstrates that the machine has, in effect, "learned;" the semantic nature of the theorem is understood by the machine.

Project Status

1. Single-machine, C++ prototype demonstration of technology complete.

   1.1 Propagation of geometry information from ProEngineer to LAPIN through PIA technology demonstrated. Performance improvement from weeks to about an hour.

2. Migration of the architecture to Common Object Request Broker Architecture (CORBA) implementation complete.

   2.1 Not just a CORBA interface to PIA; PIA is completely implemented as CORBA interfaces, some 1,100+ so far.

   2.2 First CORBA-served/PIA application wrapper, LAPIN, at hand.

3. Commercialization planning begun.

   3.1 Three Software Use Agreements in place: Emergent Technologies (LIFT), Tal-Cut Company, Entara Technologies Group.

   3.2 New activity with PGL Global/Thomas Register being explored.

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

CORBA Migration Benefits

1. Allows the architecture to extend into a collective of trusted servers and server clusters.
2. Server clusters provide high reliability/high availability, power-scaled services; remote, hot-site capabilities.
3. Allows the extension of data spaces to a practical infinity.
4. Multiple persistent storage options -- redundant, offsite, proprietary.
5. Allows multiple simultaneous consumers of served information.
6. Inherent parallelism recognized and implemented.
7. Allows cross-language consumer capabilities; a Java GUI may be made to access a C++ server.
8. Allows the services of an application to be provided without the necessity of releasing the proprietary, capital-asset code to those receiving those services; software maintenance reduction.

Tentative Commercialization Plan

Greatest advantage to the government comes from having the entire private sector talking the same integration language. To get that....

1. Core integration technologies released as open-source freeware.
   - 1.1 Commercial packaging effort needed; installation wizard.
   - 1.2 Need relationship with widely-accepted open-source site; sourceforge.net.
   - 1.3 Need to get someone interested in giving the official OK.
2. Initial applications through NASA/private sector partnerships such as Teska, Entara Technologies Group, PGL Group/Thomas Register.
3. Expected revenue streams for the private sector.
   - 3.1 Consulting, training, service delivery.
   - 3.2 Ancillary software products: workbenches, clients, operational suites, plug-and-play software, commercially-served capabilities, and the like.
   - 3.3 PIA wrapped instruments: the plug-and-play laboratory.
4. Challenge: PIA contains strong RSA encryption; Wassenaar Arrangement.

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

Project Integration Architecture Summary

1. PIA has significant potential for radically altering the way we do business.

2. Key technology components have been demonstrated in a prototype form.

3. The technology can be applied to any field: science, engineering, quality control, pharmaceuticals and rapid drug discovery, business facilitation, finance, tranportation, and many more.

4. Migration of the technology to a CORBA-implemented form is complete. Demonstration of the first wrapped application is at hand.

5. The comprehensive and exhaustive analysis of entire complex systems (launch systems, advance air transport vehicles) is enabled.

6. Such analysis can be assembled by automatic means.

7. Planning for commercialization has begun.

Backup Slides

Basic Object Primer

1. Objects are defined in classes, but are worked with as individual instances of their class.

2. Objects usually have functionality and data. Data may be shared among all the instances of a class or may be individual to each instance. Functionality is shared by all instances.

3. Classes may be derived from classes, inheriting the attributes of the parent class.

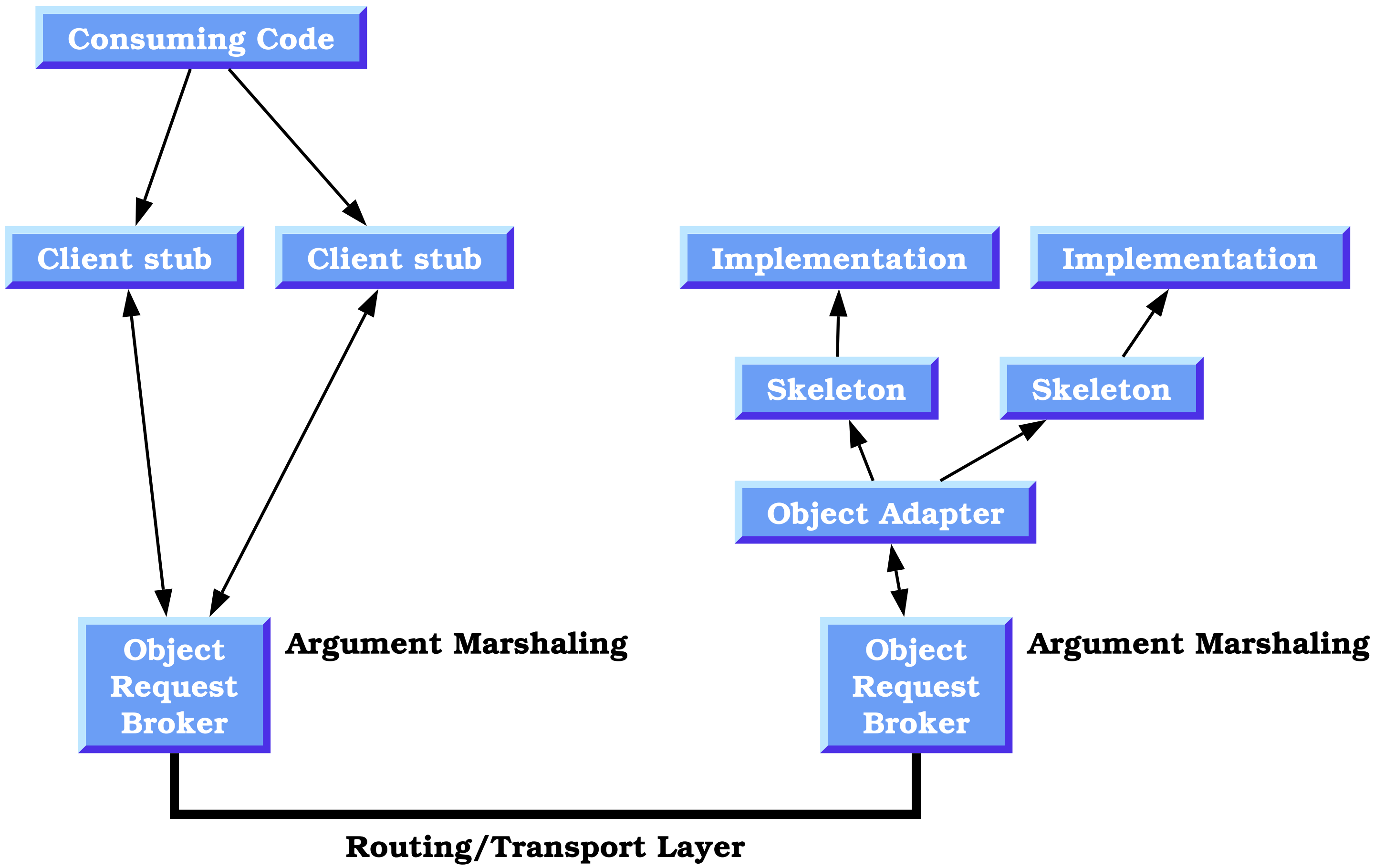
4. Derived classes may add to, override, alter, extend, turn off, or otherwise mangle inherited functionality.

5. The correct functionality is obtained without regard to whether or not the program "knew" the exact kind of object in use.

6. Usually, the first thing provided is a way to find out the kind of any given object.

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191



CORBA at a Glance

National Aeronautics and
Space Administration

John H. Glenn Research Center
Lewis Field
Cleveland, Ohio
44135-3191

Unclassified

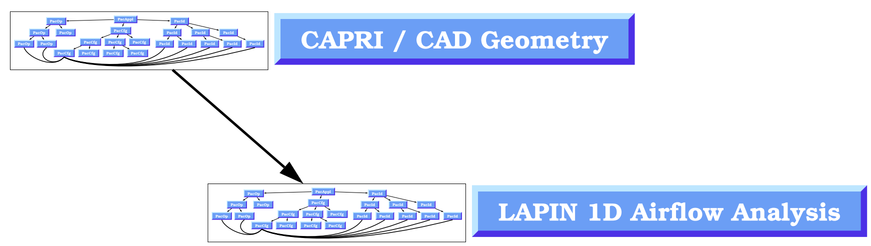
Single-Machine, C++ Prototype Status

A demonstration of self-revelation, semantic infusion through class derivation, and automated information propagation by kind has been achieved in a single-machine, C++ prototype environment. A speed-up from several weeks to about an hour has been demonstrated.

1. A PIA wrapper to the PTC/ProEngineer CAD environment using the Computational Analysis Programming Interface (CAPRI, MIT/Haimes) technology has been developed.

2. A PIA wrapper for the Large Perturbation Inlet Analysis (LAPIN) CFD code has been developed.

3. The two application wrappers have been formed into an application graph and automatic propagation of the GRC RBCC geometry information from the CAD environment to the CFD code has been demonstrated.



RBCC